

# Option Pricing with Modular Neural Networks\*

Nikola Gradojevic<sup>†</sup>

Ramazan Gençay<sup>‡</sup>

Dragan Kukolj<sup>§</sup>

January 2007

## Abstract

This paper applies a non-parametric modular neural network (MNN) model to price the S&P-500 European call options. The modules are based on time to maturity and moneyness of the options. The option price function of interest is homogenous of degree one with respect to the underlying index price and the strike price. We find that modularity improves the generalization properties of standard feedforward neural network option pricing models (with and without the homogeneity hint), relative to the Black-Scholes model.

Keywords: Option Pricing; Modular Neural Networks; Non-parametric Methods.

JEL No: C45; G12

---

\*We thank the editor, Eric Renault and two anonymous referees for their useful comments and suggestions.

<sup>†</sup>Faculty of Business Administration, Lakehead University, 955 Oliver Road, Thunder Bay, Ontario, P7B 5E1, Canada. Email: [ngradoje@lakeheadu.ca](mailto:ngradoje@lakeheadu.ca)

<sup>‡</sup>Department of Economics, Simon Fraser University, 8888 University Drive, Burnaby, British Columbia, V5A 1S6, Canada. Email: [gencay@sfu.ca](mailto:gencay@sfu.ca)

<sup>§</sup>Faculty of Engineering, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia. Email: [dragan.kukolj@MicronasNIT.com](mailto:dragan.kukolj@MicronasNIT.com)

# 1. Introduction

Since the seminal work by Black and Scholes (1973), option pricing has become the focus of scholarly attention. Most of the research efforts have attempted to relax some of the restrictive assumptions underlying the Black-Scholes model such as the normality of the log-returns or constant volatility, as there is strong empirical evidence suggesting that these assumptions are not appropriate (e.g., Bakshi *et al.* (1997)). One such example of the misspecifications of the Black-Scholes model is its substantial inaccuracy related to the pricing of the deep out-of-the-money options (Gençay and Altay-Salih (2003)). For these options, it was found that the Black-Scholes prices overestimate market prices while feedforward neural network (NN) models provide a superior pricing performance. The success of NN option pricing models was also documented in Hutchinson *et al.* (1994), Qi and Maddala (1996), Liu (1996), Garcia and Gençay (2000), and Gençay and Qi (2001).

A number of alternative approaches to account for the misspecifications of the Black-Scholes model have been proposed.<sup>1</sup> Models specified within the framework of non-constant (or stochastic) volatility were suggested by Hull and White (1987), Scott (1997), Duffie *et al.* (2000), Bakshi *et al.* (1997), Bates (2000), Pan (2002), Das and Sundaram (1999), Eraker (2004) and Chernov and Ghysels (2000). Although stochastic volatility and stochastic interest rate models are inherently misspecified, in practice they improve upon the Black-Scholes model in out-of-sample pricing and hedging exercises (Bakshi *et al.* (1997)). However, the pricing improvements are not robust and exhibit biases for the deepest out-of-the-money call options. Deterministic volatility models have also been considered (Dupire (1994), Rubinstein (1994)), but Dumas *et al.* (1998) reported that the same performance could be achieved by applying an implied volatility smoothing procedure on the Black-Scholes model. Augmenting the underlying price process can produce many different variants of the Black-Scholes model. For example, one of the most popular variants involves fitting a Poisson jump process to the stock price data (e.g., Bates (2000), Pan (2002)).

Relaxing the normality assumption of the Black-Scholes model led to parametric (Melick and Thomas (1997), Lim *et al.* (1998)) and semi-parametric (Aït-Sahalia and Lo (1998)) option pricing models. Despite the theoretical appeal of parametric models stemming from their stringent parametric assumptions that allow for simpler functional forms, non-parametric models are more flexible in relaxing the distributional assumptions of the Black-Scholes model with a potential for improved out-of-sample pricing performance. Apart from the above mentioned NN models, some notable recent research contributions that follow this approach include a non-parametric American option pricing model by Broadie *et al.* (2000b) and Broadie *et al.* (2000a), NN model for the FTSE 100 Index options by Bennell and Sutcliffe (2004), Gottschling *et al.* (2000), non-parametric locally polynomial estimator of Aït-Sahalia and Duarte (2003), positive convolution estimation of

---

<sup>1</sup>See Garcia *et al.* (2004) for a recent review.

Bondarenko (2003), and non-parametric least squares method by Yatchew and Härdle (2006).

This paper builds on the option pricing models from Hutchinson *et al.* (1994) and Garcia and Gençay (2000). Instead of relying on the method of hints from Garcia and Gençay (2000), we estimate a non-parametric model called modular neural network (MNN) model. A priori, training a single NN model on heterogeneous data (options of different moneyness and time to maturity) can be problematic for several reasons. The NN could become trapped in a local minima or may generalize poorly (Ronco and Gawthrop (1995)). Moreover, the NN model could be susceptible to the “recency effect”: the NN parameters adapted unduly in favor of the most recent training data. Feldkamp and Puskorius (1998) approach the recency effect through multistream learning, a concept related to MNNs. During multistream learning, the data set is split into multiple files (streams) and, in each training cycle, the NN parameters in streams are updated independently. This procedure improves the generalization properties of the recurrent NN and is an increasing function of the number of streams. Noteworthy, multistream learning is based on training a single NN whereas an MNN contains many NNs defined across sub-sets of the data, i.e., modules. Therefore, we find room for improvements upon simple NN option pricing models that fail to account for the recency effect. We use MNN to decompose the data into modules organized with regard to moneyness and maturity and estimate each module independently. In fact, when a priori knowledge about the model is unknown, advanced MNNs utilize learning algorithms to find the optimal modular architecture (Cofino *et al.* (2004)). We show that by dividing the data into modules and estimating each module locally it is possible to obtain a more accurate global option pricing function.

To assess the potential gains in pricing accuracy, we work within the framework of Garcia and Gençay (2000). Consequently, for the S&P-500 index European call options data, we estimate the option price to strike price ratio ( $C_t/K$ ) as a function of two arguments: the stock price to strike price ratio ( $S_t/K$ ) and the time to maturity ( $\tau$ ). Our estimation results show that MNNs are superior in capturing both cross-sectional and time-series aspects of the fluctuations in the option prices. Compared to the Black-Scholes model, for all years, except 1987, the average out-of-sample accuracy gains are statistically significant and range from 68% to 91% (three modules defined by moneyness) and from 44% to 85% (three modules defined by time to maturity). For the period 1987-1994, the average improvements of the MNN model (modules defined by moneyness) relative to the feedforward NN model with the hint lie between 1.5% in 1989 and 68% in 1987. However, the pricing accuracy of the MNN model based on the time to maturity module selection criterion is in most cases inferior to the NN model with the hint. When both of the criteria are applied simultaneously (i.e., nine modules are constructed) the MNN model provides even smaller average out-of-sample pricing errors that are consistently below the ones for the rival models. Therefore, our main empirical result is that we are able to achieve generalization improvements of the option pricing function beyond those provided by the technique of hints. We hope that our results will

foster future research on the applications of MNNs in other areas of finance and economics.

The remainder of the paper is organized as follows. Section 2 introduces the MNN model and explains its advantages over a single feedforward NN model. In section 3 we describe the data. The results of our out-of-sample pricing exercises are presented in section 4. Section 5 concludes.

## 2. Modular neural network design

### 2.1 Modular neural network estimator

This paper relies on modularity as a basic model design principle to estimate the option pricing function non-parametrically. Among many other authors, Happel and Murre (1994), Ronco and Gawthrop (1995), and Cofino *et al.* (2004) find that complex problems can be tackled by decomposing them into a number of sub-tasks. One can understand modularity as the notion of local computation on a specific task domain (or module). Each module is represented by a single feedforward NN model. Generalization is achieved through further interaction (cooperation or competition) with other modules. This concept, also called the “divide-and-conquer” method, can be very useful for function approximation in the presence of heterogeneous data. For instance, an extremely volatile region of data may be followed by a relatively stable data sequence. It can be very difficult for a single NN to extract both processes together and generalize the whole function. Sometimes referred to as “spatial crosstalk”, this problem results in a slow convergence or learning failure (Jacobs *et al.* (1991)).

The modular structure of an MNN reduces the complexity of a single NN and enhances its functional approximation capabilities. Due to their considerable flexibility and robustness to specification errors, MNNs can surpass the limitations of single NN models, or, more specifically, be able to control for the spatial crosstalk and recency effects.

To explain the concept of an MNN in the option pricing context, we will assume that a function of two variables  $c_t$  ( $c_t = \phi(x_{1t}, x_{2t}) + \varepsilon_t$ ) is driven by different functions defined over known domains of  $x_{1t}$  and  $x_{2t}$ , where  $t$  denotes the time index. Namely, function  $\phi(x_{1t}, x_{2t})$  can be written as a linear combination of functions  $c_{1t}, c_{1t}, \dots, c_{Mt}$  as follows (Jang *et al.* (1997)):

$$\phi(x_{1t}, x_{2t}) = \omega_1 c_{1t}(x_{1t}, x_{2t}) + \dots + \omega_M c_{Mt}(x_{1t}, x_{2t}) = \sum_{k=1}^M \omega_k c_{kt}(x_{1t}, x_{2t}). \quad (1)$$

This function can be approximated by an MNN (Figure 1). A meaningful decomposition of the function  $\phi(x_{1t}, x_{2t})$  would be to approximate it locally by  $M$  modules  $c_{1t}, c_{1t}, \dots, c_{Mt}$  and then to sum the output of each module, while applying the appropriate weights  $\omega_1, \omega_2, \dots, \omega_M$ . The combination weights  $\omega_k$  ( $k = 1, \dots, M$ ) can be estimated using a “softmax” method or Gaussian

mixture models. McCullagh and Nelder (1989) and Bridle (1989) propose a “softmax” activation function of each weight  $\omega_k$  as given by:

$$\omega_k = \frac{\exp(c_{kt})}{\sum_{j=1}^M \exp(c_{jt})}; \quad \sum_{k=1}^M \omega_k = 1 \quad (2)$$

Gaussian mixture models view each module as a parametric (Gaussian) distribution for local approximation of non-linear mappings (Haykin (1999)). The entire data set is then modeled as a mixture of these distributions. Put differently, the mixture density of the data is represented by a probability-weighted sum of the densities for the modules.

It is worthwhile to note that the above approaches are not explored in this paper. Instead, we specify modules (and weights) using the moneyness ( $S_t/K$ ) and time to maturity ( $\tau$ ) criteria (details are explained later in this section, after we define the option pricing function).

The modules are feedforward (backpropagation) NNs and are in general estimated as:

$$c_{kt} = g_k \left( \beta_{k0} + \sum_{j=1}^{q_k} \beta_{kj} \psi_k \left( \alpha_{kj0} + \sum_{i=1}^s \alpha_{kij} x_{it} \right) \right), \quad k = 1, \dots, M. \quad (3)$$

In our case,  $s=2$  ( $i=1,2$ ) and  $q_k$  is the number of hidden nodes for the  $k^{th}$  module. The single hidden and the output layers of the modules are characterized by two flexible classes of non-linearities:  $\psi_k$  and  $g_k$ , respectively. The backpropagation learning algorithm requires continuous differentiable non-linearities. The types used in this paper are the sigmoid logistic or hyperbolic tangent functions in the hidden layer, and the linear function in the output layer.  $\alpha_{kij}$  and  $\beta_{kj}$  denote appropriate connection weights between the adjacent layers for the modules. Subscripts 0 for  $\alpha$  and  $\beta$  stand for NN biases.

**[Insert Figure 1 about here]**

We define our option pricing formula as in Hutchinson *et al.* (1994) and Garcia and Gençay (2000):

$$C_t = \phi(S_t, K, \tau) \quad (4)$$

where  $C_t$  is the call option price,  $S_t$  is the price of the underlying asset,  $K$  is the strike price and  $\tau$  is the time to maturity. Assuming the homogeneity of degree one of the pricing function  $\phi$  with respect to  $S_t$  and  $K$ , one can write the option pricing function as follows:

$$\underbrace{\frac{C_t}{K}}_{c_t} = \phi \left( \underbrace{\frac{S_t}{K}}_{x_1}, 1, \underbrace{\tau}_{x_2} \right) = \phi(x_1, x_2). \quad (5)$$

The pricing function  $\phi$  is first approximated by an MNN with three modules determined by the time to maturity cutoff points as follows:  $\tau < 0.1$  (short term),  $0.1 \leq \tau \leq 0.2$  (medium term) and  $\tau > 0.2$  (long term).<sup>2</sup> Thus, the pricing function is decomposed into three separate non-linearities. To keep the out-of-sample prediction as simple as possible, we do not rely on any inter-module interaction. More precisely, the modules are trained independently on the data for the three option types and during prediction, based on the value of  $x_2$ , only one is active (i.e., when  $\omega_i = 1$ , all  $\omega_k = 0$ ;  $k \neq i$ ,  $k = 1, 2, 3$ ).

This empirical exercise is followed by selecting modules according to the moneyness criterion as follows:  $(S_t/K) < 0.97$  (out-of-the-money),  $0.97 \leq (S_t/K) \leq 1.05$  (near-the-money) and  $(S_t/K) > 1.05$  (in-the-money). Now,  $x_1$ 's from the prediction part of the sample determine which of the three modules is used to estimate  $c_t$ . Finally, both criteria are applied at the same time and out-of-sample predictions are estimated from an MNN with nine modules. Specifically,  $\omega_i$ 's for all nine modules are specified in the following fashion: when  $\omega_i = 1$  ( $i = 1, \dots, 9$ ),  $\omega_k = 0$  ( $k \neq i$ ,  $k = 1, \dots, 9$ ). Table 1 describes these modules specified over different ranges of  $\tau$  and  $(S_t/K)$  that are used to estimate functions  $c_{kt}$  ( $k = 1, \dots, 9$ ).

**[Insert Table 1 about here]**

Our primary goal is to improve upon the Garcia and Gençay (2000)'s model with the hint and also show that MNNs offer significant advantages over simple feedforward neural network models. The choice of the number of modules, first, follows the logic of categorizing options based on maturity (short term, medium term, long term), then, moneyness (out-of-the-money, near-the-money, in-the-money), and, finally, using both criteria. The breakpoints of the categories are from Table 4 of Garcia and Gençay (2000). Therefore, we use a “rule of thumb” approach to choosing both the number of modules and the cut-off points. In section 4, we discuss possible methods for selecting the optimal number of modules and breakpoints.

## 2.2 Alternative non/semi-parametric estimators

Non-parametric models such as MNNs make minimal assumptions about the shape or the properties of the underlying price process. In addition, non-parametric pricing functions are convenient for estimating state price densities (SPDs) and the “Greeks.” For instance, the MNN estimator can be directly differentiated numerically (or analytically) to extract the SPD or option delta. This section will describe in fairly general terms the relationship between artificial neural networks (NNs) that are building blocks for MNNs and some alternative non- and semi-parametric estimators.

---

<sup>2</sup>The units for  $\tau$  are the number of days to maturity divided by the number of days in a year (365 or 366).

Similar to linear regression models, NNs map a set of explanatory variables ( $x_{it}$ , for example,  $i = 1, 2$ ) into a dependent variable ( $c_t$ ). The major difference between NNs and other non-parametric estimators is that NNs are based on a layered structure where each layer transforms  $x_{it}$ 's using special functions known as sigmoid logistic or hyperbolic tangent functions (see equation 3). The number of layers and the structure of each layer have to be determined before estimation, like the bandwidth in kernel regression methodologies. In the case of NNs, this is typically performed through a cross-validation method on the validation part of the data (Garcia and Gençay (2000)). Then, the parameters of an NN are estimated by minimizing the loss function defined as the sum of squared differences between the observed  $c_t$  and the one predicted by an NN. The goal of this paper is to estimate the parameters for each NN of the MNN option pricing function.

A popular non-parametric alternative to NNs to estimate the pricing function is kernel regression (Aït-Sahalia and Lo (1998)). In contrast to NNs, where observational errors are reduced by averaging the data based on a recursive error minimization procedure, for kernel regression this is achieved by local averaging (smoothing). The kernel method estimates the price based on the weighted sum of the information from the in-sample data. More specifically, given some vector  $(x_{1t}; x_{2t})$ , to estimate the price, more weight is assigned to the observations at locations that are closer to the vector. As our option pricing model has two regressors, the appropriate kernel pricing function would be constructed as a product of two univariate kernels. The usual trade-off between smoothness and goodness-of-fit is achieved by the choice of the bandwidth of the kernel function. As previously suggested, in an NN, this trade-off is controlled by the structure and the number of the layers, or, more generally, by the number of modules in an MNN. Aït-Sahalia and Lo (1998) show that the kernel estimation errors for the SPD and option delta lie within one per cent of the theoretical values. The out-of-sample pricing performance of their model is also impressive and superior over the NN model. It is noteworthy that the kernel pricing function from Aït-Sahalia and Lo (1998) that is used for forecasting is semi-parametric. It is informed by dividend yield,  $\tau$ ,  $K$ ,  $S_t$ , risk-free interest rate, and non-parametrically estimated volatility. In contrast, the variables in the MNN model are the ratio of the asset price to the strike price and the time to maturity.<sup>3</sup>

Other estimators that can be employed for option pricing include nearest-neighbour estimators, splines, average derivative estimators, local polynomial regression, and orthogonal series expansion. For example, the nearest-neighbours method is based on an assumption that geometric patterns in the past of the time series, similar by some measure (e.g., Euclidean distance) to the currently observed variables, can be used for forecasting (see, e.g., Yakowitz (1987)). This set of "related" observations of independent variables is called the nearest neighbours. The smoothing constant analogous to the kernel bandwidth determines the number of nearest neighbours to be considered

---

<sup>3</sup>We also compared the forecast performance of the semi-parametric model to the MNN model. The results can be found in section 4 (Table 3).

in the estimation. In general, all of the above methods are concerned with averaging or smoothing the data in a more or less sophisticated fashion while controlling for the trade-off between bias and variance.

### 3. The data and assessment of prediction performance

The data are daily S&P-500 index European call option prices taken from the Chicago Board Options Exchange. For each available complete year, over the period from January 1987 to October 1994, options across different strike prices and maturities are considered. Being one of the deepest and the most liquid option markets in the United States, the S&P-500 index option market is sufficiently close to the theoretical setting of the Black-Scholes model. This also makes our study directly comparable to Garcia and Gençay (2000).

Options with zero volume are not used in the estimation. As in Garcia and Gençay (2000), for the NN estimation purposes, the data for each year are divided into three parts: first two quarters (training set), third quarter (validation set) and fourth quarter (out-of-sample, testing set). This produced the following non-overlapping sub-samples:

- **1987:** Training sample: 3610, Validation sample: 2010, Testing sample: 2239
- **1988:** Training sample: 3434, Validation sample: 1642, Testing sample: 1479
- **1989:** Training sample: 3052, Validation sample: 1565, Testing sample: 1515
- **1990:** Training sample: 3610, Validation sample: 2010, Testing sample: 2239
- **1991:** Training sample: 4481, Validation sample: 1922, Testing sample: 2061
- **1992:** Training sample: 4374, Validation sample: 1922, Testing sample: 1848
- **1993:** Training sample: 4214, Validation sample: 1973, Testing sample: 2030

The optimal NN architecture for each module is determined from the out-of-sample performance on the validation set with respect to the mean-squared prediction error (MSPE). To keep the complexity of the NNs close to Garcia and Gençay (2000), the modules are single hidden layer NNs with either sigmoid logistic or hyperbolic tangent activation functions. Hence, the search for an optimal NN architecture involves specifying the number of hidden nodes (we choose from 1-15 hidden nodes) and their activation functions. The parameters are estimated using the standard Levenberg-Marquardt algorithm. Overfitting is prevented by early stopping, i.e., stopping the training process when the validation set error starts to increase. To control for possible sensitivity



of the NNs to the initial parameter values, the training is performed from ten different random seeds and the average MSPE values are reported. The predictive performance on the testing part of the sample is finally assessed with the MSPE criterion.

The out-of-sample pricing performance of an MNN model is compared to the Black-Scholes model.<sup>4</sup> The Black-Scholes call prices ( $C_t$ ) are computed using the standard formula:

$$C_t = S_t N(d) - K e^{-r\tau} N(d - \sigma\sqrt{\tau}) \quad \text{where} \quad d = \frac{\ln(S_t/K) + (r + 0.5\sigma^2)\tau}{\sigma\sqrt{\tau}} \quad (6)$$

where  $N$  is the cumulative normal distribution,  $S_t$  is the price of the underlying asset,  $K$  is the strike price,  $\tau$  is the time to maturity,  $r$  is the risk-free interest rate, and  $\sigma$  is the volatility of the underlying assets continuously-compounded returns estimated from the last sixty days preceding the first day of the last quarter. The risk-free rate is approximated by the monthly yield of the U.S. Treasury bills.

The statistical significance of the prediction performance is assessed with the Diebold-Mariano statistic (Diebold and Mariano (1995)) that is distributed as standard normal in large samples.<sup>5</sup> When the calculated values of the Diebold-Mariano statistic are large and positive, the MNN model is viewed as being able to significantly improve upon the Black-Scholes model.

## 4. Results

Table 2 compares the out-of-sample pricing performance of the NN model with the hint, the MNN model with three modules and the Black-Scholes model in terms of the MSPE performance measure. The MNN model is estimated ten times from ten different sets of starting values and the average MSPEs are reported along with the average number of hidden layer units in modules. The third column in Table 2 reveals that when the modules are selected based on the moneyness criterion, the MNN model outperforms the NN model with the hint for all years. Furthermore, the same architecture of the MNN model consistently outperforms the Black-Scholes model, except in 1987. The pricing improvements are statistically significant according to the Diebold-Mariano statistic. Garcia and Gençay (2000) also find 1987 to be problematic, but show that when the third quarter (validation) data are included in the training set, the performance of the NN model with the hint becomes superior to the Black-Scholes model. We will show later that by increasing the complexity

---

<sup>4</sup>This will also shed more light on the magnitude of our model's forecasting improvements, relative to the model with the hint from Garcia and Gençay (2000).

<sup>5</sup>West (1996) argued that this statistic is potentially unreliable. More precisely, forecast errors can be both serially and contemporaneously correlated. We compute the test statistic as the average of the forecast error differences for the ten estimations of the optimal NN architectures.

of the MNN model we can outperform the Black-Scholes model without using the validation data for training.

**[Insert Table 2 about here]**

The fourth column in Table 2 reveals that when the modules are selected based on time to maturity, the MNN model does not perform well. In almost all of the years, the MSPEs are greater than the ones for the model with the hint, but still below the ones for the Black-Scholes model at the 5% significance level. Again, this is not the case with 1987. We conjecture that the complexity of the option pricing function is such that it requires more (or different) modules.

**[Insert Table 3 about here]**

To give more flexibility to the MNN model we employ both module selection criteria over nine modules. Table 3 shows that this model is more accurate than both the Black-Scholes and the NN with the hint models. This holds for all years and the improvements in the MSPE are statistically significant. The performance of the MNN model is impressive for 1987 as well. Recall that the volatility in the Black-Scholes model is calculated from the information contained in the third quarter. As opposed to the NN model with the hint, for 1987, the MNN model does not require any additional training data to ensure superior pricing performance. To illustrate the accuracy of the MNN model we plot out-of-sample predictions of  $C_t/K$  and the actual data across the moneyness dimension for 1990 and 1993 (Figure 2: Panels A and B). As can be seen in the panels, the estimates follow the actual prices very closely, especially in 1993, when the average MSPE is much smaller than the one in 1990.

**[Insert Figure 2 about here]**

We also compare the MNN model with other alternative approaches such as the kernel regression (Aït-Sahalia and Lo (1998)) and the parametric mixture model (Melick and Thomas (1997)). The third column of Table 3 indicates that, for various choices of bandwidth, the kernel model could not improve upon the MNN model in any of the years, except in 1987.<sup>6</sup> To implement the mixture method, we find the European option price as the discounted value of the probability distribution above the strike price. Following Melick and Thomas (1997), we use the mixtures of two and three

---

<sup>6</sup>We followed a semi-parametric kernel approach from Aït-Sahalia and Lo (1998) and, first, estimated implied volatilities based on futures price, the exercise price and time to maturity using the Nadaraya-Watson kernel estimator. Then, call option prices were computed from the estimated volatility function via the Black-Scholes formula. The differences between our estimation and the original approach are that we do not use the dividend yield and the rolling estimation procedure. Nonetheless, the kernel model depends on more variables than the MNN model which gives it significant informational advantage.

lognormal distributions over option series consisting of call options with different strike prices but same maturity. We find that the out-of-sample forecast performance of the mixture model over the years is either inferior to the kernel model or similar to the Black-Scholes model.<sup>7</sup> Aït-Sahalia and Duarte (2003) show that by constraining the pricing function the pricing error can be reduced by about 25-50% for lower bandwidths (and much smaller sample sizes). However, a 50% reduction in the pricing error of the kernel pricing function would not be sufficient to outperform the MNN model. Furthermore, Daglish (2003) demonstrates that the shape-constrained kernel produces very modest pricing accuracy gains for large data sets such as ours. Consequently, we do not include this enhancement of the nonparametric kernel into our set of rival models.

Based on these results, we conclude that the modularity feature of the MNN model plays a vital role in outperforming the rival models. This also accords with some other studies that focused on pricing of certain types of options (i.e., deep out-of-the-money options) and found that such partitioning of the data increases the pricing accuracy of the NN models (Gençay and Altay-Salih (2003), Bennell and Sutcliffe (2004)). However, as shown for the time to maturity-constructed modules, one has to be cautious when using available domain knowledge to identify modules. Finally, we find the performance improvements closely linked to the number of modules in an MNN.

Increasing the number of modules, however, adds to the complexity of the estimated pricing function. As previously explained, our choice for the number of modules and the breakpoints for the modules is arbitrary, but sensible. Nevertheless, at this global level, we do not control for the usual trade-off between smoothness and goodness-of-fit in calibration of non-parametric methods, but only optimize the performance of individual modules. Clearly, moving from three to nine modules improves the accuracy of the MNN model. By observing the average pricing errors of the individual modules of the three- and nine-module MNNs, we first attempt to pin down the cause for these improvements. Tables 4 and 5 contain average pricing errors for the modules for 1993 that are representative for all other years.

**[Insert Table 4 about here]**

**[Insert Table 5 about here]**

Table 4 suggests that when the modules are selected by time to maturity, the pricing performance is poor for two modules (short term and long term options). The alternative three-module MNN has problems pricing in-the-money options (i.e., only one module performs poorly) and this seems to be the major reason for its superior performance over the first MNN variant. The nine-module MNN confirms that in-the-money options are relatively difficult to price for all maturities

---

<sup>7</sup>For brevity, these results have not been reported in Table 3, but can be available from the authors upon request.

(Table 5). Also, it appears that the information that comes from the moneyiness criteria compensates for the time horizon effects. In other words, the modules perform roughly equally well for different time to maturity ranges.

We further increase the number of modules from nine to sixteen, i.e., add one more range along each dimension. This is carried out by dividing the middle moneyiness range (0.97-1.05) into two ranges around 1 (0.97-1, 1-1.05) and the middle maturity range (0.1-0.2) into [0.1-0.15] and (0.15-0.2]. The average MSPE for 1993 as a result of this exercise slightly increases from 0.0546e-04 to 0.0656e-04. For all other years the average error for the sixteen-module MNN is about 10-25% larger. This indicates that the optimal number of modules is greater or equal to nine and less than sixteen. As searching for an optimal MNN architecture is beyond the scope of this paper, for future research purposes, we recommend that the optimal number of modules and the cut-off points could be determined either by observing the performance on the validation data (third quarter of each year) or by utilizing some techniques from the statistical learning theory, as suggested in the last section. More precisely, one could select the optimal number of modules ( $M$ ) by using a cross-validation approach on the validation part of the data. Yet, this procedure does not identify the module boundaries and we will offer a possible iteration method for finding them. Starting from some initial boundaries for a given  $M$ , in each iteration the “goodness” of current partition can be determined using the Davies - Bouldin ( $DB$ ) index (Davies and Bouldin (1979)):

$$DB = \frac{1}{M} \sum_{i=1}^M R_i, \quad R_i = \max(R_{ij}), \quad j = 1, \dots, M; j \neq i \quad (7)$$

where  $R_{ij} = (S_i + S_j)/D_{ij}$  is a similarity measure between modules  $i$  and  $j$ ,  $S_i$  is a dispersion measure of the  $i^{th}$  module calculated as the average Euclidean distance of the data points in module  $i$  to its center, and  $D_{ij}$  is a module dissimilarity measure (distance between the centers of the module pairs). Finally, the partition that produces the minimum  $DB$  is considered to be optimal.

Our data set is such that on any given day there are about 25 to 50 options of different strike price and maturity. Hence, the sample size required for training the MNN model is not large, unlike the sample size for the kernel methodology. Aït-Sahalia and Duarte (2003) note that their semi-parametric kernel model requires a relatively large data set to be empirically effective. This feature of our model makes it very suitable for practical applications.

One of the relevant practical uses of non-parametric option pricing is the estimation of the SPDs, which allows practitioners to price OTC derivatives consistently with the prices of exchange traded options. To obtain the SPD and the options delta, we find analytical derivatives of equation (4) in a standard way:

$$\hat{\Delta}_t = \frac{\partial \hat{C}(S_t, K, \tau)}{\partial S_t} \quad (8)$$

$$\hat{f}_t^* = e^{r_{t,\tau}\tau} \left[ \frac{\partial^2 \hat{C}(S_t, K, \tau)}{\partial K^2} \right]_{|K=S_T} \quad (9)$$

Even though our pricing function is very accurate out-of-sample, an unconstrained estimator such as the MNN could produce large errors for the derivatives. Initially, we observed a wiggly and sometimes negative SPD surface. This violates general properties of a density function and, in turn, violates the no-arbitrage principle (Aït-Sahalia and Duarte (2003)). In addition, the impact of the violations could be more acute and, for instance, lead to trading on false arbitrage signals. This is also explained in Garcia *et al.* (2004) and Garcia and Gençay (2000) who observed similar deficiencies in terms of derivatives of a nonparametric pricing function.

To alleviate for this problem we followed Dugas *et al.* (2001) and imposed the *softplus* activation function in the hidden layer of each NN:

$$\psi(Z) = \log(1 + e^Z); \quad \psi'(Z) = \frac{e^Z}{1 + e^Z}; \quad \psi''(Z) = \frac{e^Z}{(1 + e^Z)^2}. \quad (10)$$

This function and its first two derivatives are always positive. Moreover, its first derivative is the sigmoid logistic function that is constrained between 0 and 1, like a probability distribution function. The second derivative has the following properties:  $\lim_{Z \rightarrow -\infty} \psi''(Z) = 0$  and  $\lim_{Z \rightarrow +\infty} \psi''(Z) = 0$ . Therefore, it behaves like a probability density function. Dugas *et al.* (2001) show that the *softplus* function satisfies the universal approximator property and, thus, can replace the sigmoid logistic function in a backpropagation NN.

In panels A and B of Figure 2 we plot the average values for deltas and SPDs for 1993 estimated by a constrained MNN model.<sup>8</sup> The averages are received for the sample time to maturity that ranges from 1 to 360 days. The other variables that enter the estimation of deltas and SPDs are set to their average values for 1993. Despite some limitations in terms of smoothness, the non-parametric SPD surface still exhibits excess skewness and kurtosis and which is also found by Aït-Sahalia and Lo (1998). More importantly, it is bell-shaped and always positive. With regard to the estimated delta, minor discrepancies from the theoretical values are observed in panel B of Figure 2, but the shape is reasonable and the values lie between 0 and 1. Note that the wiggleness can be observed mostly near the cutoff points between the modules and could be a direct consequence of the modular complexity of the MNN. For a single NN model, Hutchinson *et al.* (1994) also find that deltas estimated by the NN model from do not significantly deviate from their theoretical counterparts.

---

<sup>8</sup>This did not deteriorate the MSPE of the model that remained roughly 0.05e-04.

[Insert Figure 3 about here]

One way to deal with unconstrained non-parametric estimators is to impose monotonicity and convexity (Yatchew and Härdle (2006)) by employing well-behaved functional forms for the NNs in the modules. In general, the term “well-behaved” refers to a non-negative function that integrates to unity and we showed that imposing such shape constraints is a potentially promising future research avenue. To specify a valid density function, Gottschling *et al.* (2000) propose an alternative method based on an NN approximation and the logarithm of the inverse Box - Cox transformation. The simplest algorithm for imposing constraints on non-parametric techniques is used in Aït-Sahalia and Duarte (2003): before any estimation is performed, the data are repeatedly transformed until they are monotonic.

## 5. Conclusions

The goal of this paper is to investigate whether a non-linear MNN model can be a successful option pricing tool. To assess the performance of the MNN model, we study S&P-500 European call options over the 1987-1993 period. Rather than relying on the method of hints, we estimate pricing functions using the modularity feature of the MNNs. This approach yields advantages not only over the Black-Scholes model, but also over the NN model with the hint. When the MNN model utilized three modules selected with regard to the moneyness criterion, the performance of the MNN model is consistently superior to the model with the hint in all years. However, an alternative variant of the MNN model based on the time to maturity criterion performs poorly. Moreover, both three-module MNN models are inferior to the Black-Scholes model in 1987. Extending the MNN model to nine modules reduced the pricing error to below the one for the model with the hint and significantly improved upon the Black-Scholes model for the whole sample.

We conclude that the MNN option pricing model, or more generally, the notion of modularity is a promising future research direction in options pricing. Moreover, selection of modules and data partitioning based on more advanced methodologies from the statistical learning theory such as fuzzy clustering (Jang *et al.* (1997)) or genetic algorithms (Cofino *et al.* (2004)) can complement the current model. This can be of assistance, for instance, in improving the time to maturity criterion in the MNN model and also in gaining better insight into its potential pitfalls. More importantly, the success of our approach suggests that MNNs may be used to capture and explain highly volatile and elusive data-generating processes.

	$(S_t/K) < 0.97$	$0.97 \leq (S_t/K) \leq 1.05$	$(S_t/K) > 1.05$
$\tau < 0.1$	$c_{1t}$ (module 1)	$c_{2t}$ (module 2)	$c_{3t}$ (module 3)
$0.1 \leq \tau \leq 0.2$	$c_{4t}$ (module 4)	$c_{5t}$ (module 5)	$c_{6t}$ (module 6)
$\tau > 0.2$	$c_{7t}$ (module 7)	$c_{8t}$ (module 8)	$c_{9t}$ (module 9)

Table 1: MODULE SELECTION CRITERIA FOR THE MNN OPTION PRICING MODEL WITH NINE MODULES.

Notes: The options are categorized based on maturity (short term, medium term, long term) and moneyness (out-of-the-money, near-the-money, in-the-money). When both criteria are applied at the same time, the out-of-sample predictions are estimated from an MNN with nine modules. Specifically,  $\omega_i$ 's for all nine modules are specified in the following fashion: when  $\omega_i = 1$  ( $i = 1, \dots, 9$ ),  $\omega_k = 0$  ( $k \neq i, k = 1, \dots, 9$ ).

Year	Statistic	ANN with hint	MNN model ( $S_t/K$ )	MNN model ( $\tau$ )	BS model
1987	$\overline{MSPE}$	16.7	5.44 [9]	6.8 [10]	4.38
	$\sigma$	9.51	2.30	4.02	
	DM		-4.23	-7.75	
1988	$\overline{MSPE}$	0.7114	0.6681 [5]	1.1592 [6]	2.07
	$\sigma$	0.0429	0.0134	0.4304	
	DM		8.36	3.30	
1989	$\overline{MSPE}$	0.4138	0.4076 [5]	0.5093 [6]	1.42
	$\sigma$	0.0068	0.0014	0.1152	
	DM		9.61	4.65	
1990	$\overline{MSPE}$	0.6761	0.5759 [9]	0.6639 [6]	2.62
	$\sigma$	0.0763	0.0597	0.1742	
	DM		17.23	14.06	
1991	$\overline{MSPE}$	0.3498	0.3401 [5]	0.3727 [6]	1.73
	$\sigma$	0.0148	0.0226	0.0244	
	DM		7.17	6.38	
1992	$\overline{MSPE}$	0.1511	0.1326 [8]	0.2044 [10]	1.36
	$\sigma$	0.0115	0.0123	0.0534	
	DM		6.60	3.97	
1993	$\overline{MSPE}$	0.1054	0.0665 [10]	0.1442 [11]	0.74
	$\sigma$	0.0222	0.0068	0.0637	
	DM		10.01	2.65	

Table 2: PREDICTION PERFORMANCE OF THE MNN OPTION PRICING MODEL WITH THREE MODULES.

Notes: The first row of this table contains the out-of-sample average mean-squared prediction errors ( $\overline{MSPE}$ ) of the following option pricing models: Garcia and Gençay (2000)'s feedforward neural network model with the hint (ANN with hint), modular neural network (MNN) model with the moneyness ( $S_t/K$ ) module selection criterion, MNN model with the time to maturity ( $\tau$ ) module selection criterion and the Black-Scholes model (BS model). The average number of hidden layer nodes in MNN modules is reported in the square brackets. The average MSPEs for the MNN models have been obtained as averages across ten different random training seeds.  $\sigma$  is the standard deviation of the ten MSPEs estimated from ten different random seeds. DM denotes the Diebold and Mariano (1995) test statistic. This test is used to assess the statistical significance of the MNN model's forecast gains relative to the BS model. All  $\overline{MSPE}$  and  $\sigma$  figures have been multiplied by  $10^4$ .



	ANN with hint	MNN model	AL	BS Model	DM	$\sigma$
$\overline{MSPE}$						
1987	16.7	4.1204 [4]	3.3665	4.38	3.04	1.0376
1988	0.7114	0.6107 [5]	2.3784	2.07	10.88	0.0372
1989	0.4138	0.4008 [5]	1.5854	1.42	9.99	0.0038
1990	0.6761	0.5579 [5]	1.3428	2.62	26.89	0.0334
1991	0.3498	0.3293 [4]	0.8351	1.73	8.30	0.0023
1992	0.1511	0.1285 [4]	0.2756	1.36	6.85	0.0073
1993	0.1054	0.0546 [3]	0.4263	0.74	11.42	0.0062

Table 3: PREDICTION PERFORMANCE OF THE MNN OPTION PRICING MODEL WITH NINE MODULES.

Notes: This table reports the out-of-sample average mean-squared prediction errors ( $\overline{MSPE}$ ) of the following option pricing models: Garcia and Gençay (2000)'s feedforward neural network model with the hint (ANN with hint), modular neural network (MNN) model with nine modules, Aït-Sahalia and Lo (1998)'s semi-parametric estimator (AL), and the Black-Scholes model (BS model). The average number of hidden layer nodes in MNN modules is reported in the square brackets. The average MSPEs for the MNN models have been obtained as averages across ten different random training seeds.  $\sigma$  is the standard deviation of the ten MSPEs estimated from ten different random seeds. DM denotes the Diebold and Mariano (1995) test statistic. This test is used to assess the statistical significance of the MNN models forecast gains relative to the BS model. All  $\overline{MSPE}$  and  $\sigma$  figures have been multiplied by  $10^4$ .

	$(S_t/K) < 0.97$ or $\tau < 0.1$	$0.97 \leq (S_t/K) \leq 1.05$ or $0.1 \leq \tau \leq 0.2$	$(S_t/K) > 1.05$ or $\tau > 0.2$
MNN model ( $S_t/K$ )	0.0407	0.0620	0.1884
MNN model ( $\tau$ )	0.2490	0.0731	0.1672

Table 4: AVERAGE MSPEs FOR INDIVIDUAL MODULES (1993, THREE MODULES).

Notes: This table reports the out-of-sample average mean-squared prediction errors ( $\overline{MSPE}$ ) for each module of a three-module MNN model. The figures for both versions of the model are reported: modules selected by the moneyiness criterion (first row) and modules selected by the time to maturity criterion (second row). The average MSPEs for the models have been obtained as averages across ten different random training seeds. All  $\overline{MSPE}$  figures have been multiplied by  $10^4$ .

	$(S_t/K) < 0.97$	$0.97 \leq (S_t/K) \leq 1.05$	$(S_t/K) > 1.05$
$\tau < 0.1$	0.0018	0.0407	0.12
$0.1 \leq \tau \leq 0.2$	0.0179	0.0457	0.2094
$\tau > 0.2$	0.0313	0.0595	0.2161

Table 5: AVERAGE MSPEs FOR INDIVIDUAL MODULES (1993, NINE MODULES).

Notes: This table reports the out-of-sample average mean-squared prediction errors ( $\overline{MSPE}$ ) for each module of a nine-module MNN model. The options are categorized based on two criteria that are applied simultaneously: maturity (short term, medium term, long term) and moneyness (out-of-the-money, near-the-money, in-the-money). The average MSPEs for the models have been obtained as averages across ten different random training seeds. All  $\overline{MSPE}$  figures have been multiplied by  $10^4$ .

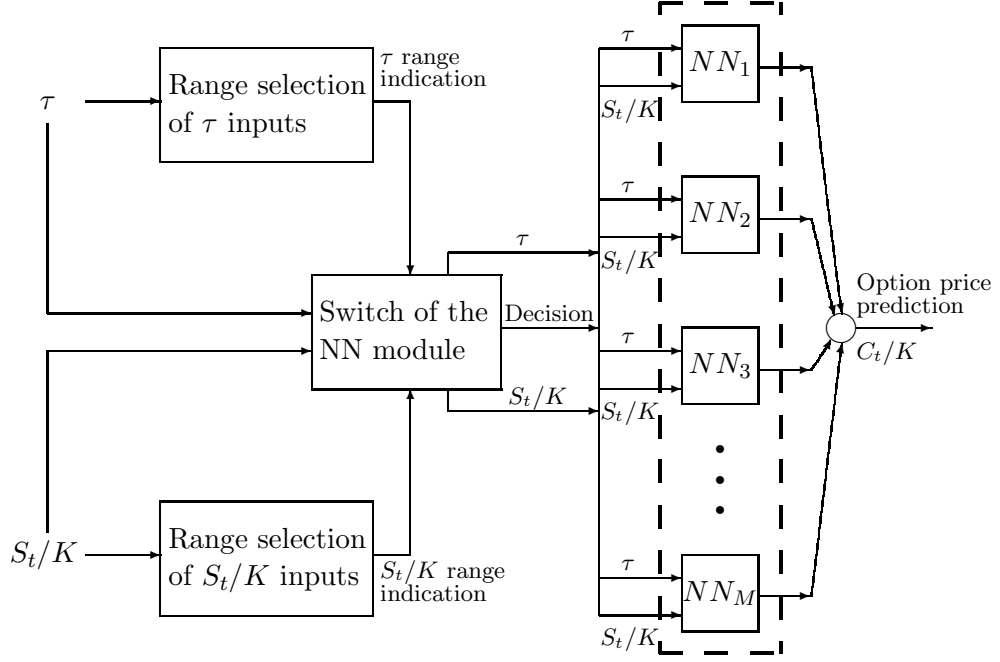
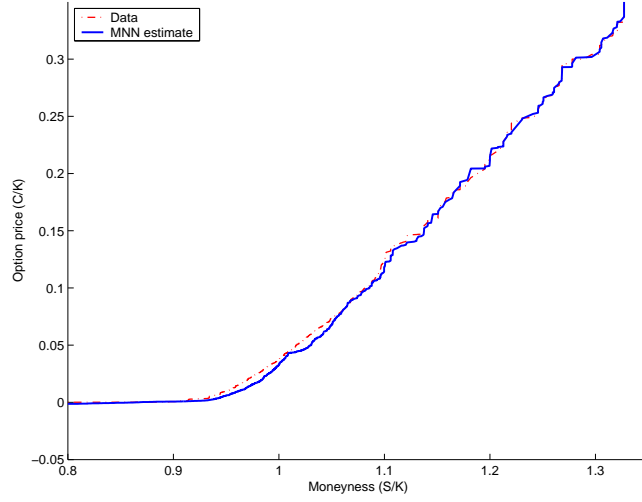
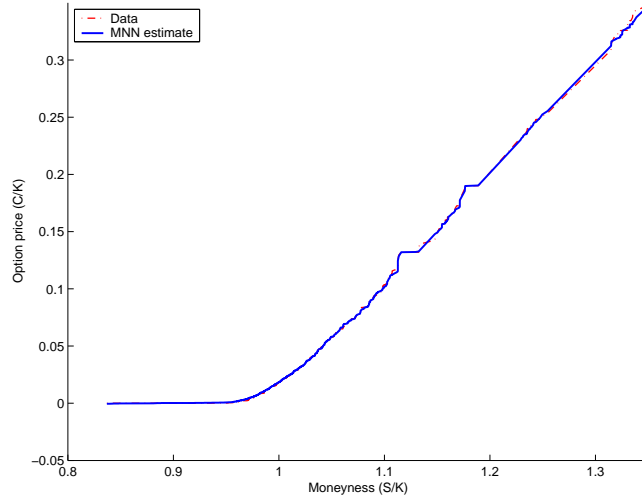


Figure 1: Modular neural network architecture.

Notes: The pricing function is decomposed into  $M$  separate non-linearities ( $NN_1, \dots, NN_M$ ) called modules. The modules are trained independently on the data for the  $M$  option types and during prediction, based on the values of  $\tau$  and  $S_t/K$ , only one NN is active (i.e., when  $\omega_i = 1$ , all  $\omega_k = 0$ ;  $k \neq i$ ,  $k = 1, \dots, M$ ). More specifically, the out-of-sample predictions ( $C_t/K$ ) are generated from nine different NN models that span the whole testing set.



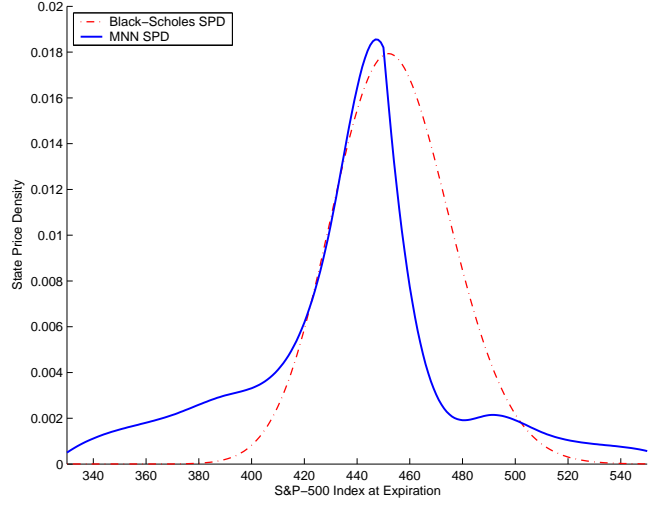
Panel A



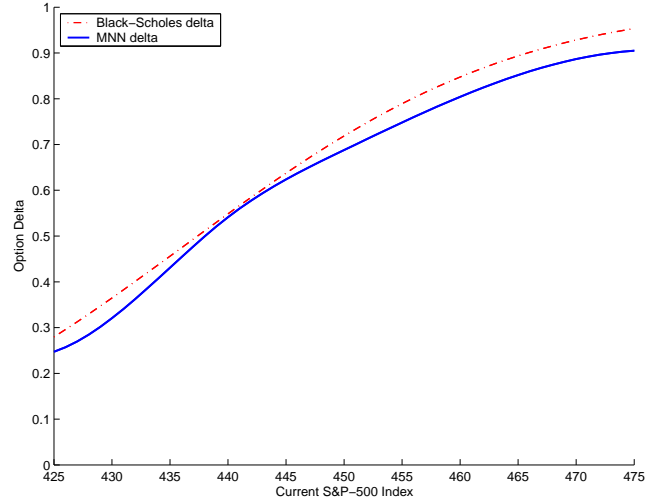
Panel B

Figure 2: Data and option prices estimated by the nine-module MNN model.

Notes: Out-of-sample predictions of  $C_t/K$  (solid line) and the actual data (dashed line) are plotted against  $S_t/K$  in Panels A (1990) and B (1993). First, the MNN model is trained using the data from the first two quarters of each year and, then, 2239 (for 1990) and 2030 (for 1993) out-of-sample estimates of  $C_t/K$  are generated. The average MSPEs for 1990 and 1993 are  $0.5579\text{e-}04$  and  $0.0546\text{e-}04$ , respectively.



Panel A



Panel B

Figure 3: The SPDs and deltas of the constrained nine-module MNN estimator for 1993.

Notes: Panel A depicts SPD estimates from the constrained nine-module MNN model (solid line) and the Black-Scholes SPDs (dashed line). To obtain the SPDs, the second derivative of the MNN estimator is evaluated analytically at  $K = S_T$ . Panel B depicts estimates of an option delta from the constrained nine-module MNN model (solid line) along with the ones from the Black-Scholes model (dashed line). The deltas are estimated as the analytical first derivatives of the MNN pricing function with respect to  $S_t$ . All other relevant variables are fixed at their sample means for 1993. The estimates of SPDs and deltas are averaged across the sample time to maturity values that range from 1 to 360 days.

## References

- Aït-Sahalia, Y. and Duarte, J. (2003). Nonparametric option pricing under shape restrictions. *Journal of Econometrics*, **116**, 9–47.
- Aït-Sahalia, Y. and Lo, A. (1998). Nonparametric estimation of state-price densities implicit in financial asset prices. *Journal of Finance*, **53**, 499–547.
- Bakshi, G., Cao, C., and Chen, Z. (1997). Empirical performance of alternative option pricing models. *Journal of Finance*, **52**(5), 2003–2049.
- Bates, D. S. (2000). Post-'87 crash fears in the s&p 500 futures option market. *Journal of Econometrics*, **94**(1/2), 181–238.
- Bennell, J. and Sutcliffe, C. (2004). Black-scholes versus artificial neural networks in pricing ftse 100 options. *International Journal of Intelligent Systems in Accounting, Finance and Management*, **12**, 243–260.
- Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, **81**, 637–659.
- Bondarenko, O. (2003). Estimation of risk-neutral densities using positive convolution approximation. *Journal of Econometrics*, **116**, 85–112.
- Bridle, J. S. (1989). *Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition*, pages 227–236. Springer-Verlag, New York.
- Broadie, M., Detemple, J., Ghysels, E., and Torr  s, O. (2000a). American options with stochastic dividends and volatility: A nonparametric investigation. *Journal of Econometrics*, **94**, 53–92.
- Broadie, M., Detemple, J., Ghysels, E., and Torr  s, O. (2000b). Nonparametric estimation of american options' exercise boundaries and call prices. *Journal of Economic Dynamics and Control*, **24**, 1829–1857.
- Chernov, M. and Ghysels, E. (2000). A study towards a unified approach to the joint estimation of objective and risk neutral measures for the purpose of options valuation. *Journal Of Financial Economics*, **56**, 407–458.
- Cofino, A. S., Guti  rrez, J., and Ivanissevich, M. (2004). Evolving modular networks with genetic algorithms: Application to nonlinear time series. *Expert Systems*, **21**(4), 208–216.
- Daglish, T. (2003). A pricing and hedging comparison of parametric and non-parametric approaches for american index options. *Journal of Financial Econometrics*, **1**(3), 327–364.
- Das, S. and Sundaram, R. (1999). Of smiles and smirks, a term structure perspective. *Journal of Financial and Quantitative Analysis*, **34**, 211–239.
- Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **1**(2), 224–227.
- Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics*, **13**, 253–263.

- Duffie, D., Pan, J., and Singleton, K. (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, **68**, 1343–1376.
- Dugas, C., Bengio, Y., Belisle, F., Nadeau, C., and Garcia, R. (2001). Incorporating second-order functional knowledge for better option pricing. *Advances in Neural Information Processing Systems*, **13**, 472–478.
- Dumas, B., Fleming, J., and Whaley, R. E. (1998). Implied volatility functions: Empirical tests. *Journal of Finance*, **53**, 2059–2106.
- Dupire, B. (1994). Pricing with a smile. *Risk*, **7**(1), 18–20.
- Eraker, B. (2004). Do stock prices and volatility jump? reconciling evidence from spot and option prices. *Journal of Finance*, **59**, 1367–1403.
- Feldkamp, L. and Puskorius, G. (1998). A signal processing framework based on dynamic neural networks with application to problems in adaptation, filtering, and classification. In *Proceedings of IEEE*, volume 86, pages 2259–2277.
- Garcia, R. and Gençay, R. (2000). Pricing and hedging derivative securities with neural networks and a homogeneity hint. *Journal of Econometrics*, **94**(1/2), 93–115.
- Garcia, R., Ghysels, E., and Renault, E. (2004). *The Econometrics of Option Pricing*. Elsevier-North Holland, Amsterdam. Forthcoming.
- Gençay, R. and Altay-Salih, A. (2003). Degree of mispricing with the black-scholes model and nonparametric cures. *Annals of Economics and Finance*, **4**, 73–101.
- Gençay, R. and Qi, M. (2001). Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping and bagging. *IEEE Transactions on Neural Networks*, **12**(4), 726–734.
- Gottschling, A., Haefke, C., and White, H. (2000). Closed form integration of artificial neural networks with some applications to finance. Technical report, University of California, San Diego. Working Paper.
- Happel, B. and Murre, J. (1994). Design and evolution of modular neural network architectures. *Neural Networks*, **7**, 985–1004.
- Haykin, S. (1999). *Neural Networks*. Prentice Hall, Inc., NJ.
- Hull, J. and White, A. (1987). The pricing of options on assets with stochastic volatilities. *Journal of Finance*, **42**, 281–300.
- Hutchinson, J. M., Lo, A. W., and Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *Journal of Finance*, **43**(3), 851–889.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, **3**, 79–87.
- Jang, J. R., Sun, C., and Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*. Prentice Hall, Inc., NJ.



- Lim, G. C., Lye, J. N., Martin, G. M., and Martin, V. L. (1998). The distribution of exchange rate returns and the pricing of currency options. *Journal of International Economics*, **45**, 351–368.
- Liu, M. (1996). Option pricing with neural networks. In S. I. Amari, L. Xu, L. W. Chan, I. King, and K. S. Leung, editors, *Progress in Neural Information Processing*, volume 2, pages 760–765. Springer-Verlag.
- McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman and Hall, London.
- Melick, W. R. and Thomas, C. P. (1997). Recovering an asset’s implied pdf from option prices: An application to crude oil during the gulf crisis. *Journal of Financial and Quantitative Analysis*, **32**(1), 91–115.
- Pan, J. (2002). The jump-risk premia implicit in options: Evidence from an integrated time series study. *Journal of Financial Economics*, **63**, 3–50.
- Qi, M. and Maddala, G. S. (1996). Option pricing using artificial neural networks: The case of s&p 500 index call options. In A. P. N. Refenes, Y. Abu-Mostafa, J. Moody, and A. Weigend, editors, *Neural Networks in Financial Engineering: Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, pages 78–91, New York. World Scientific.
- Ronco, E. and Gawthrop, P. J. (1995). Modular neural networks: A state of the art. Technical Report CSC-95026, Centre for System and Control, University of Glasgow.
- Rubinstein, M. (1994). Implied binomial trees. *Journal of Finance*, **49**(3), 771–818.
- Scott, L. (1997). Pricing stock options in a jump-diffusion model with stochastic volatility and interest rates: Applications of fourier inversion methods. *Mathematical Finance*, **7**, 413–426.
- West, K. D. (1996). Asymptotic inference about predictive ability. *Econometrica*, **64**, 1067–1084.
- Yakowitz, S. J. (1987). Nearest neighbour methods for time series analysis. *Journal of Time Series Analysis*, **8**(2), 235–247.
- Yatchew, A. and Härdle, T. (2006). Nonparametric state price density estimation using constrained least squares and the bootstrap. *Journal of Econometrics*, **133**(2), 579–599.